CS 43111/53111  STRUCTURE OF COMPILERS  3 credit hours

Instructor’s Name: Dr. Ruoming Jin

Textbook:

Course Content:
(Cross-listed with CS 53111) Techniques used to write compilers including lexical analysis, syntax analysis, syntax-directed translation, type checking, run-time environments, and intermediate code generation
Prerequisites or co-requisites: CS 35101 and 33101. Required, elective, or selected elective

Goals:
1. Understand the basics of compilers
2. Learn lexical and syntax analyses
3. Learn syntax-directed translation
4. Learn type checking
5. Get familiar with run-time environments
6. Learn intermediate code generation

Outcomes:
1. Understanding the structure of compilers
2. Be proficient with the process of compilers
3. Write programs to implement parsing functions in compilers
4. Write programs to implement type-checking functions in compilers
5. Write a compiler program

Topics to be Covered:
1. Introduction to Compilers
2. Lexical Analysis
3. Syntax Analysis
4. Top-Down Parsing
5. Bottom-Up Parsing
6. Advanced Parsing
7. Semantic Analysis
8. Type-Checking
9. Runtime Environments
10. Local Optimization
11. Global Optimization
12. Register Allocation
13. Garbage Collection
14. Code Optimization

**ABET Learning Outcomes:**

- Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
- Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
- Apply computer science theory and software development fundamentals to produce computing-based solutions.